Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Computer Science 10 (2012) 609 – 616
Procedia Computer Science

The 9th International Conference on Mobile Web Information Systems (MobiWIS)

A Multi-Criteria Approach for Web Service Discovery

Raja Ben lakhal[a], Walid Chainbi[b*]

[a] *ISG University of Tunis/SOIE, 41 Avenue de la Liberté Bouchoucha, Bardo, TUNISIA*
[b] *National Engineering School of Sousse/SOIE, Sousse -4054-, TUNISIA*

Abstract

With the growth of Web Services number and with the diversification of their types and qualities, the choice of the service that complies with user preference became a fundamental issue. The task of finding the best Web service is no longer an easy task and may end to unsatisfactory result. To resolve this problem, we adopt in this paper a Web services discovery and selection approach based on user preferences. These preferences are based on a multi-criteria approach that allows clients to assign weights values to the QoS parameters according to their needs. Consequently, the selected services will correspond to the real clients needs. In our application, we have also integrated a mechanism for semantics recommendation. This recommendation serves to guide the client preferences and to refine more the results given by the selection mechanism based on QoS parameters. Unlike other approaches, in our work, we separate between recommendation and discovery mechanism. This will give the client the freedom to choice his own scenario and to gain in terms of processing time. Our approach integrates also geospatial criteria which are of paramount importance in mobile applications.

Key words: Web Services discovery; multi-criteria approach; semantic recommendation; UDDI; QoS (Quality of Service).

1. Introduction

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Service-Oriented Computing (SOC) is a computing paradigm that utilizes Web services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments [1]. Web services

* Corresonding author. Tel: +216-73-369-500.
E-mail address: Raja.Belakhal@gmail (R. Ben Lakhal), Walid.Chainbi@gmail.com (W. Chainbi).

1877-0509 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.procs.2012.06.078

are software systems designed to expose functions that are accessible using standard Web technologies and that adhere to Web services standards and architecture [2][3].

Given the importance of solutions presented by Web Services, many standards and architectures were developed to support this technology, but these solutions were not developed sufficiently to ensure a proper Web Service selection and discovery. Therefore the client was not satisfied with the results presented by these classic methods. These shortages are present mainly in the UDDI registry structure and in the Web Services description standards languages such as WSDL description. In fact the UDDI registry allows only keywords search, whereas the standard language description of services does not take into account the semantic aspect.

So in order to satisfy the client needs, Quality of Web Service parameters were introduced. In this context, many discovery approaches were proposed including the work of Chen Zhou and al [4]. The authors proposed a new QoS parameters modeling framework. This model is an extension of DAML-S representation, it allows an object-oriented representation of the QoS quality parameters. Shuping Ran [5], propose a new Framework that extends UDDI registry structure with QoS parameters. This framework integrates a certifier that ensures the reliability of QoS parameters values.

In our work, we introduce QoS parameters which allow the user to find relevant services that correspond to his preferences and enable him also to gain in term of time by minimizing his search space. Indeed, the client is not obliged to spend a lot of time to find the suitable service and to search through different resources such as portals, search engine and so.

We consider in our work four types of QoS parameters which are: Cost, Response Time, Availability and Reliability.

- *Cost*: Cost involved in the usage of a Web service.
- *Response Time*: Time taken to send a request and receive a response.
- *Availability*: Number of successful invocation/total invocation.
- *Reliability*: Ratio of the number of error messages to total messages.

In our work, we adopt a centralized architecture that is based on Service Oriented Architecture (SOA). This architecture is composed of three main parts, namely the provider, the client and the UDDI registry. In this architecture, the UDDI registry is of a great importance because it's the point of conjunction between all clients and providers. We adapt the UDDI structure with the addition of the QoS parameters. We use the OWL-S description language augmented by the QoS parameters to describe all our services.

In our application, the discovery mechanism is conducted by the client that has the role to introduce the weights values according to his preferences. This is done for each QoS parameter. In each execution, the client may choose at least one parameter and assign to it a weight value. As a result, a list of sorted services is displayed in accordance to the calculated scores. These scores are computed based on a maximization function that takes into account the weights introduced by the clients and the parameters values selected from the UDDI registry. Our application integrates a recommendation mechanism providing the client with useful hints concerning a better Web service selection. This mechanism includes four kinds of semantic parameters which are: geo-spatial position, the Website type where the service will be deployed, the Web service security level and the infrastructure in which the Web Service will be used such as PC (Personnel Computer), iPhone, etc.

The reminder of this paper is organized as follows. Section 2 deals with the related work that treated the task of the best Web service selection. In section 3, we introduce our work starting with Web Services

description and publication mechanisms. Each service is described with the OWL-S language enhanced by QoS parameters [6]. The mechanism of mapping between an OWL-S document and UDDI is also discussed. Section 4 deals with selection approaches that we have adopted in order to choose the best Web service. In section 5, we deal with the recommendation mechanism and focuses on geospatial recommendation, that we have implemented within our Web service discovery framework.

2. Related work

In order to solve classic Web Service discovery methods problems and gaps, a lot of efforts have been resulting in various approaches which differ in the used methods of selection and in the adopted architecture. However, the majority were in agreement with the principle of the introduction of (QoS) parameters and in the consideration of customer preferences in Web service selection.

Chen, Song, Zhang and Miao [7] proposed a Web service selection approach based on the similarity concept between services inputs and outputs. These similarities were calculated according to the associative rules algorithm result. The similarity concept enables to measure the degree of correspondence between a requester and a provider service. Given a client request and a service provider, the highest degree of similarity measured for those two propositions is considered the best service for a given client preference.

Al-Masri and Mahmoud in [8] proposed another approach that introduces the principal of the Broker. The Broker is a framework that collects and manages various Web Services that are distributed throughout the Web. It allows also the QoS attributes management by considering continuous update of their values. The best Web Service search mechanism is based on a function that enables to find the relevant Web Service according to a given user and a given parameter weight. In fact the service which is with the greatest function value will be considered the best service for a particular client.

Other approaches are based on the Web services reputations. In these approaches, an agent uses the collected reputation ratings from clients who had previous experience and usage with Web services. For example, Xu et al proposed a selection approach based on reputations and users feedbacks which are reflected by scores [9]. These scores are introduced by clients according to their degree of satisfaction against a given service quality. A database was introduced to store all scores for each service. These scores will be used for future comparisons with new client preferences.

Other studies discussed the recommendation mechanism based on user's experiences [10, 11]. For example, in [10] the proposed approach adopts a special type of recommendation in distributed environments. The idea is to search a service across different registries by trying each time to find the group of registries that match more the client request query. Then based on the registries group that was selected, another type of recommendation called "User-based characterization technique recommendation" is applied to find the most appropriate register corresponding to the same client request. This latter recommendation is based on the user experience.

Sun et al [12] proposed in their paper a selection and recommendation approach based on case-based reasoning (CBR). Such a recommendation is based on the user experience and history of selected services scenarios.

In the aforementioned studies, the recommendation part is generally treated based on the scores submitted by users and taking into account the user experiences in the selection process. However, this approach can be ineffective in the case where the client request is not processed before or in the case where the context

of the selected service does not match client desires. Moreover, in some studies there is no separation between the selection and the recommendation process (e.g., [13, 14]).

In our work, we deal with these problems by implementing an application that takes into account user preferences denoted by weight values and by using comparisons operators. Moreover, in order to address the problem of semantics lack, we have adopted a semantic and a GPS recommendation approach that take into account the real client preferences. We have also separated between selection and recommendation to give the customer the freedom, either to adopt the results produced by the two Web service discovery approaches that we have presented in our work, or to adopt the result returned by the recommendation part. Actually, by making this separation, the client is no longer obliged to follow one scenario that may be costly for him in terms of waiting time, but we give him the freedom to choose his own scenario. The main contribution of our work is the definition and the implementation of a global Framework for the discovery. We also use a recommendation mechanism that is running in the background. This recommendation is integrated into the discovery process while letting the customer the possibility to make his final choice.

3. Web Service description and publication

In this section, we adopt the OWL-S description language to describe our services. In fact, for each Web service two types of OWL files have been defined. The first file contains all the elements that define a service which are service *profile*, the service *process* and the service *grounding*. The service profile defines the service Web with three types of information which is: service provider information, operation service information and characteristics of service information. In the service process description part, services are presented according to three kinds of process, namely; atomic process, simple process and composite process. This part answers the question; *how* a service works. The last part is the service Grounding. It presents the concrete specification of the service as opposed to Service profile and Service process which present the abstract specification. The Grounding service contains details of the protocol to use, address networks and formats of incoming and outgoing messages. The second file is the extension of the first one. It contains the QoS and the semantics parameter values. In order to create our files, we have used the Protégé software and the OWL-S API editor. We have defined three different instances of a service file description that belongs to different providers. In our application, for example, we use an email validation service and we define three description instances service witch correspond respectively to *MailBoxValidation* description, *EmailValidation* description and *ValidateEmail* description. Figure1 correspond to the *MailBoxValidation* description that takes into account the QoS and semantic parameters. We have also used the JDOM API to extract the QoS and the semantics parameters values from the OWL-S description that we have already made. We have designed also a Web interface that allows all providers to introduce additional information about the service, such as service name, service address, service description, provider name, provider address. Once everything is defined the client can publish his service on the UDDI registry. We have adopted as implementation for the UDDI registry a MYSQL database that contains tables that represents the real UDDI structure (Tmodel, Tmodel_Category, etc…).

4. Web Service Selection

In our work, user preferences are expressed by two multi-criteria approaches. The first approach enables the user to enter a weight value for each chosen QoS parameter. The user may choose at least one parameter and at most all of them, according to his preferences. For example, if the user is interested by the *Availability* parameters, he has to assign weight values that are between 0.1 and 0.9. The other parameters take automatically 0. If the client is not interested by a given parameter such as the *Response Time* parameter, which reflects his desire to reduce the waiting time, so he should assign the value 0 to the

weight that correspond to this parameter. Hence, the services that have the greatest *Response Time* parameter values will not be chosen among the best Web Services. However, if the client is not interested to have a big or small *Response Time* value, he can then assign a weight value that belong to the interval [0.1, 0.9]. Weights introduced by the client are taken into account in the maximization function. This function is equal to the sum of product between the weight value and the corresponding parameter value for each service. Due to the fact that QoS parameters vary in units, for example, the *Availability* parameter is the *%* and the *Cost* is *USD*, QoS metrics must be normalized to be able to compute the maximization function. Normalization enables a unified distribution of parameter measures that have different units. In addition, normalization prevent that parameters which have the biggest measures, impact more the maximization function result. In order to calculate the maximization function, we need before to compute the normalized values for each parameter.

Given a list of candidate Web Services ($SW_{1...}$ SW_m). Each service is described in terms of parameter values list ($P_{1...}$ P_n) and by considering the set of weight values for each parameter ($W_{1...}$ W_n). In our case n=1...4.

Let N be the values sum for each QoS parameter.

$$N(n) = \sum_{k=1}^{m} P_k \quad (1)$$

Where P_k, represent a parameter values for a given service. The normalized value is equal to the following equation:

$$e_i = P_i / N(n) \quad // i=1..n \quad (2)$$

```
<ServiceQuality rdf:ID="ServiceQuality_20">
  <profile:sParameter>
    <ServiceQualityInfo rdf:ID="ServiceQualityInfo_21">
      <Reliability rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.7</Reliability>
      <Cost rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >19</Cost>
      <Availability rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >0.9</Availability>
      <ResponseTime rdf:datatype="http://www.w3.org/2001/XMLSchema#time"
      >00:05:33</ResponseTime>
    </ServiceQualityInfo>
  </profile:sParameter>
</ServiceQuality>
 <profile:serviceParameter>
    <GPS rdf:ID="GPS_24">
      <profile:sParameter>
        <GPSInfo rdf:ID="GPSInfo_25">
          <longitudeGPS xml:lang="string">10.16596</longitudeGPS>
          <latitudeGPS xml:lang="float">36.81881</latitudeGPS>
        </GPSInfo>
      </profile:sParameter>
    </GPS>
  </profile:serviceParameter>
 <SemanticQuality1 rdf:ID="SemanticQuality_22">
      <profile:sParameter>
        <SemanticQualityInfo rdf:ID="SemanticQualityInfo_23">
          <security xml:lang="string">high</security>
          <sitetype xml:lang="string">e-commerce</sitetype>
          <country xml:lang="string">USA</country>
          <platforme xml:lang="string">iPhone</platforme>
        </SemanticQualityInfo>
      </profile:sParameter>
    </SemanticQuality1>
```

Fig.1. MailBoxValidation example of QoS and semantics parameters description.

The maximization function value will be equal to:

$$F(SW_j) = \sum_{i=1}^{n} e_i * W_i \quad // j=1...m \quad (3)$$

The service j that has the highest F function value will be the best Web Service for a given client.

In our application, we have adopted two kinds of Web services multi-criteria selection approaches: The first one is based on the weights given by the client and the other approach is based on comparison

operators. In the first approach, the client is responsible for giving weight values to parameters that he considers the most relevant. While in the second approach, the client expresses his choice by entering quality of services parameters values preceded by comparison operators (<,>, =). Indeed, the client can specify more than one criterion at once, as is illustrated by figure 3. As the UDDI registry may contain more than one type of service published by different providers, we have introduced a search filter to allow the user to have as results only the services that match his thematic of search. In both approaches, we have developed an selection algorithm that enables to extract the QoS parameters values from the UDDI registry and to use them either to make comparisons with the values introduced by the client (case of the second approach) or to use them in the maximization approach that we have defined above(case of the first approach). Also, we have adopted a simple ranking algorithm to classify services according to their scores. As shown in figure 2, the selection result will be a ranked services list which presents the service name, the last service update, the service score and a textual description that will be used for further improvements to our application. In fact, this information can be stored and used for processing further comparisons with new client preferences.

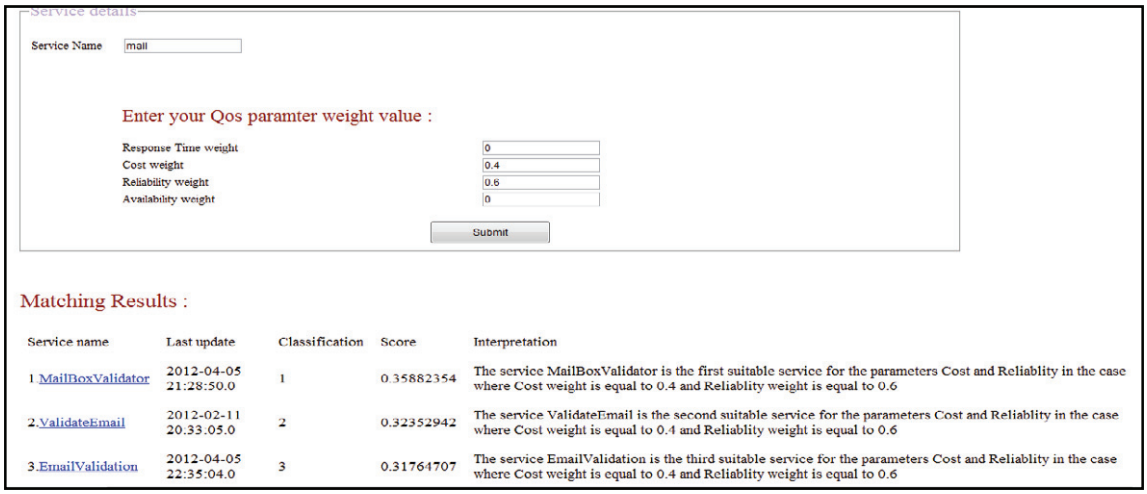

Fig. 2. A multi-criteria approach for Web service selection

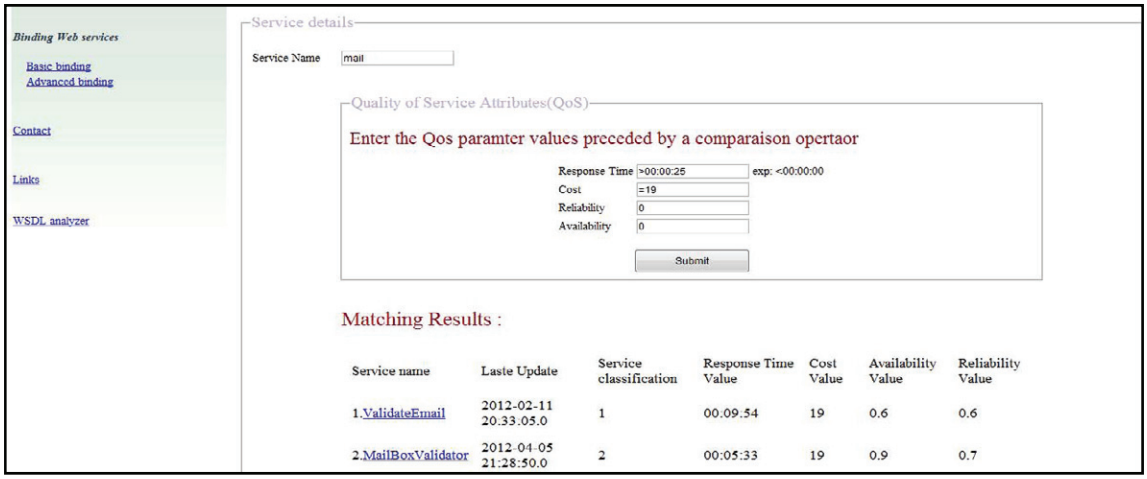

Fig. 3. The best Web service selection based on comparisons operators

5. Semantics Recommendation

Semantics recommendation that we have adopted does not deal with non-functional parameters such as QoS parameters, but serves to express the semantics preferences of the customers. These preferences are expressed by semantics criteria including the geospatial criteria. These criteria may interest clients that want to have a service that takes into account, for instance, the jurisdiction of a country. So the addition of this criterion in the recommendation process helps the client to find a service that presents the same legal rules (relevant to the Web Service Semantics) as those presented in his country of residence. This recommendation process run in background and generate as result a set of services that matches with semantic users preferences. These preferences help customers to discover new criteria that can improve their search process of the best service. In contrast, when the client considers only the selection based QoS approach, he could not be aware that the service he has chosen may not be the best one that will satisfy other criteria which are not specified by him. For example, in the case where the client looks for an e-commerce service that is located in his country of residence, the selection based QoS approach, will not enable him to find the suitable service. So, in this case the introduction of new semantics parameters that reflects the right geospatial provider position becomes necessary. In our application, we have adopted other kinds of criteria including the geospatial position, the platform used, and the security level. The latter criterion is important mainly in the case where the application in which the Web Service will be deployed needs a high level of security. In our work, we have separated between selection and recommendation. Actually, this separation enables to gain in terms of the execution time, especially in the case when the client is satisfied by the selection process result based on QoS parameters and when he is not interested by any other criteria that the application integrates. The recommendation concerns all the selected services by the selection process. This recommendation return to the user a list of services ranked according to their name, the semantic parameters values and the classification range for each one. The user can also benefit from the result of the GPS recommendation developed in our application. This recommendation is based on the comparisons between provider's and actual client positions. The client position is automatically generated. It is defined with two parameters namely *longitude* and *latitude.* These parameters are generated automatically every time the client is connected to the GPS recommendation interface. The GPS provider's information is extracted from services descriptions that were selected by the discovery process based on the QoS parameters. The service which has the closest *longitude* and *latitude* values to those of the client will be considered the best service according to the GPS criterion.

6. Conclusion

In this paper, we have defined two types of discovery Web service approaches. One based on quality of services parameters, and the other is based on the comparisons operators. We have also introduced the principle of semantics recommendation which includes other criteria that are not taken into account by a QoS based selection approach. Such criteria include the GPS position which is of paramount importance in mobile applications. The proposed selection approaches as well as the semantics selection mechanisms have been implemented resulting in a Web services discovery framework. In future work, we envision to enrich the publication side with a veracity checker of the QoS parameters values. Concerning the Web Services discovery part, improvements are underway in order to enhance our discovery framework by considering other parameters that may affect the client choice.

References

[1] Munindar P.Singh and Michael N.Huhns. Service-Oriented Computing: Semantics, Processes, Agents. John Wiley & Sons, Ltd, West Sussex, England; 2005.

[2] E. Newcomer. Understanding Web Services-XML, WSDL, SOAP and UDDI, Addison Wesley Professional-Independent Technology Guides series, Reading, Massachusetts; 2002.

[3] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. Web Service Architecture; 2004.

[4] Chen Zhou, Liang-Tien Chia and Bu-Sung Lee. Semantics in Service Discovery and QoS Measurement . IEEE Educational Activities Department Piscataway, NJ, USA; 2005, pp. 29-34.

[5] Shuping Ran. A Framework for Discovering Web Services with Desired Quality of Services Attributes. International Conference on Web Services; 2003, pp. 208-213.

[6] Gustavo Fortes Tondello. The QoS-MO Ontology for Semantic QoS Modeling. 2008 ACM symposium on Applied computing; 2008, pp. 2336-2340.

[7] Li Chen, Zi-lin Song, Ying Zhang and Zhuang Miao. A Method of Web Service Matchmaking Based on Snippets. AISS: Advances in Information Sciences and Service Sciences; 2011, pp. 250 –258.

[8] Eyhab Al-Masri and Qusay H. Mahmoud. A Broker for Universal Access to Web Services.2009 Seventh Annual Communications Networks and Services Research Conference; 2009.

[9] Ziqiang Xu, Patrick Martin, Wendy Powley and Farhana Zulkernine. Reputation-Enhanced QoS-based Web Services Discovery; 2007, p. 249–256.

[10] Mohamed Sellami, Samir Tata, Zakaria Maamar and Bruno Defude. A Recommender System for Web Services Discovery in a Distributed Registry Environment. ICIW '09 Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services; 2009, p. 418-423.

[11] Freddy Lécué. Combining Collaborative Filtering and Semantic Content-based Approaches to Recommend Web Services. Semantic Computing (ICSC), 2010 IEEE Fourth International Conference; 2010, p.200 – 205.

[12] Zhaohao Sun, Jun Han and Dianfu Ma. A Unified CBR Approach for Web Services Discovery, Composition and Recommendation. International Conference on Machine Learning and Computing, IACSIT ICMLC 2009, Mercure Perth Hotel, Perth, Western Australia; 2009.

[13] M. Brian Blake and Michael F.Nowlan. A Web Service Recommender System Using Enhanced Syntactical Matching, Web Services, 2007. ICWS2007. IEEE International Conference; 2007, pp. 575 – 582.

[14] WEN Jun-hao, Zheng Chang and LI Peng. Research on the Model of Dynamic and Mixed QoS Personalized Semantic Web Service Recommendation. JCIT: Journal of Convergence Information Technology; 2011, pp. 41-48.